Combining PPO and Evolutionary Strategies for Better Policy Optimization

Jennifer She
Computer Science, Stanford University
jenshe@stanford.edu

Objective

- Propose and implement hybrid policy optimization methods inspired by Proximal Policy Optimization (PPO) and Natural Evolutionary Strategies (ES) in order to leverage their individual strengths
- Compare hybrid methods against PPO and ES in two OpenAI environments: CartPole and BipedalWalker

Background

Under the reinforcement learning (RL) framework, the goal of policy optimization is to find a policy \( \pi_\theta : S \times A \rightarrow [0, 1] \) defining \( \mathbb{P}(a_t = a|s_t = s) \) that maximizes the expected return

\[
J(\theta) = \mathbb{E}_{s \sim p(s)} [\mathbb{R}_t(\theta)],
\]

PPO updates \( \pi_\theta \) via an approximation of \( \nabla_{\theta} J(\theta) \)

- pro: it uses gradient information to guide its updates, which helps it to zero-in on potential solutions
- con: it may get stuck at a local optima as a result

ES parameters \( \theta \) with

\[
\Theta = \theta + \sigma \varepsilon \sim \mathcal{N}(0, I)
\]

which it updates by sampling \( \{\theta(1), \ldots, \theta(k)\} \) weighted by their return

\[
1 / k \sigma^2 \sum_{i=1}^{k} [\Sigma_{[1:p(i)]} r_t(\pi_\theta)^{\frac{1}{2}}]^{1/2}
\]

- pro: it incorporates stochasticity in the space of \( \theta \) for better exploration of \( \pi_\theta \)
- con: it treats the RL problem as a black-box

The goal is then:

To build hybrid methods that both leverage gradient information and are stochastic in \( \theta \)

Methods

ES-PPO

- Sample \( \theta(i) \) as in ES, but instead, run PPO with these as initializations to obtain \( \theta(i)^p \)
- Update \( \pi_\theta \) by (1) with modified perturbations

\[
\phi' = \frac{1}{\sigma} (\phi'' - \theta)
\]

MAX-PPO

- Run ES/PPO as above but directly set \( \bar{\theta} \) to \( \theta(i)^p \) with the highest return

\[
\text{argmax} \sum r_t(\pi_\theta)^{\frac{1}{2}}
\]

ALT-PPO

- Run ES every \( j \) PPO iterations

We compare these methods to ES and PPO

Environments

CartPole-v0 (CP)

- \( S \subset \mathbb{R}^2, A = \{0, 1\} \)
- Objective: Move cart to keep pole upright
- Rewards: +1 every timestep for a max of 200
- Termination: Pole falls / cart goes off screen or episode reaches max of 200 timesteps

BipedalWalker-v2 (BW)

- \( S \subset \mathbb{R}^4, A = [-1, 1]^4 \)
- Objective: Maneuver walker to right-most side of environment (target) without falling
- Rewards: +1 for moving forward, for a total of 300 on agent reaching target; -100 for falling
- Termination: Walker reaches target or falls

Architecture Details

ES

\[
\pi_\theta(a|s) = 1[a = f_\theta(s)]
\]

where \( f_\theta \) is a fully-connected neural network

- FC(dims(s) \times 100) + ReLU
- FC(100 \times dim(a))
- Sigmoid + 1\] \( \{ \text{CP} \} \) or Tanh \( \{ \text{BW} \} \)

PPO/Hybrids

\[
\begin{align*}
\pi_\theta(a|s) &\sim \text{Bernoulli}(g_\theta(s)) \quad \text{(CP)} \\
\pi_\theta(a|s) &\sim \mathcal{N}(g_\theta(s), \sigma) \quad \text{(BW)}
\end{align*}
\]

where \( g_\theta \) is a fully-connected neural network

- FC(dims(s) \times 100) + ReLU
- FC(100 \times dim(a) + ReLU
- FC(100 \times dim(a))
- Sigmoid \( \{ \text{CP} \} \) or Tanh \( \{ \text{BW} \} \)

Results

<table>
<thead>
<tr>
<th>Environment</th>
<th>Return</th>
<th>Training Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>ES</td>
<td>200.0</td>
<td>60.59</td>
</tr>
<tr>
<td>PPO</td>
<td>200.0</td>
<td>53.74</td>
</tr>
<tr>
<td>ES-PPO</td>
<td>200.0</td>
<td>515.03</td>
</tr>
<tr>
<td>MAX-PPO</td>
<td>200.0</td>
<td>363.52</td>
</tr>
<tr>
<td>ALT-PPO</td>
<td>200.0</td>
<td>131.24</td>
</tr>
</tbody>
</table>

Table: Final results from CP averaged across 5 trials

Discussion

- PPO and ES performed well on both tasks
- PPO: Training instability (BW) likely a result of reusing samples from \( \pi_{\theta_{\text{init}}} \)
- ES: Evaluating \( \theta(i) \) is slow without leveraging large-scale parallel compute \( \rightarrow \) extending ES-PPO and MAX-PPO from ES exponentiated this problem, and forced us to choose max sample size \( k = 5 \) for BW
- ES-PPO: PPO calls may drive \( \theta(i)^{\text{opt}} \) far from \( \bar{\theta} \); thus a weighted average of returns at \( \theta(i)^{\text{opt}} \) may no longer be a good predictor of return at weighted average of \( \theta(i)^{\text{opt}} \) \( \rightarrow \) misleading update signals
- MAX-PPO: Mitigates averaging problem of ES-PPO but may lead away from a good solution when all neighbouring \( \theta(i)^{\text{opt}} \) have low returns \( \rightarrow \) high variance
- ALT-PPO: Mitigates high computation cost of ES-PPO and MAX-PPO but its stochasticity may lead away from a good solution when neighbour \( \theta(i)^{\text{opt}} \) have low (but different) returns

Future Directions

- Investigate trade-offs in sample efficiency and variance in the case of PPO
- Investigate ways to leverage high-compute in the case of ES-PPO and MAX-PPO
- Investigate stochasticity with adaptive variance (using gradient information) to avoid moving away from good solutions
- Investigate more complex environments where ES and PPO fail

Acknowledgements

We thank Mario Srouji (TA) for the project idea, and help during the project.